

FINAL PROJECT

JAVA Application Minikube deployment

DAY -6

The image displays two screenshots related to a Java application deployment project.

The top screenshot shows the Jenkins web interface for a pipeline named "java application". The pipeline is in a "Completed" state. The "Stage View" shows the following stages and their durations:

Stage	SCM	Build	Build Docker Image	Push Docker Image	test
#15	711ms	1s	310ms	14s	
#14	746ms	1s	313ms	16s	392ms
#13	1s	1s	563ms	17s	1s
#11	724ms	1s	299ms	14s	
#10	1s	1s	305ms	22s	

The bottom screenshot shows a terminal window with the following commands and output:

```
navin@ITP-CC16-42:~$ minikube start
minikube v1.35.0 on Ubuntu 24.04 (amd64)
Using the docker driver based on existing profile
Starting "minikube" primary control-plane node in "minikube" cluster
Pulling base image v0.0.46 ...
Restarting existing docker container for "minikube" ...
Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
Verifying Kubernetes components...
  * Using image gcr.io/k8s-minikube/storage-provisioner:v5
  * Enabled addons: storage-provisioner, default-storageclass
Docker kubect1 is now configured to use "minikube" cluster and "default" namespace by default
navin@ITP-CC16-42:~$ minikube service my-service
NAMESPACE   NAME       TARGET PORT   URL
default     my-service  9000          http://192.168.49.2:30002
Starting tunnel for service my-service.
NAMESPACE   NAME       TARGET PORT   URL
default     my-service  9000          http://127.0.0.1:44875
Opening service default/my-service in default browser...
http://127.0.0.1:44875
! Because you are using a Docker driver on linux, the terminal needs to be open to run it.
^C Stopping tunnel for service my-service.
navin@ITP-CC16-42:~$ minikube service my-service
NAMESPACE   NAME       TARGET PORT   URL
default     my-service  9000          http://192.168.49.2:30002
Starting tunnel for service my-service.
NAMESPACE   NAME       TARGET PORT   URL
default     my-service  9000          http://127.0.0.1:45581
Opening service default/my-service in default browser...
http://127.0.0.1:45581
! Because you are using a Docker driver on linux, the terminal needs to be open to run it.
```

Pipeline script

Pipelinescript-

Jenkins JAVA

Application

```
pipeline {
```

```
    agent any
```

```
    stages { stage('scm'){
```

```
        steps{
```

```
            gitbranch:" https://github.com/NavinkumarD/Guvi-Tasks-DevOps.git"
```

```
        }
```

```
    }
```

```
    stage('builb
```

```
        -clean'){
```

```
        steps {
```

```
            sh"mvn clean"
```

```
        }
```

```
    }
```

```
    stage('build-
```

```
        validate'){
```

```
        steps {
```

```
            sh"mvn validate"
```

```
}
```

```
}
```

```
stage('buil
```

```
  d-com'){
```

```
    steps {
```

```
      sh"mvn compile"
```

```
    }
```

```
}
```

```
stage('buil
```

```
  d-test'){
```

```
    steps {
```

```
      sh"mvn test"
```

```
    }
```

```
}
```

```
stage('build-
```

```
  install'){
```

```
    steps {
```

```
      sh"mvn package"
```

```
    }
```

```
}
```

```
stage('buildtoima
```

```
  ges'){
```

```
    steps {
```

```
      script{
```

```

        sh'docker build-t.'
    }
}
}
stage('pushtohub
    '{ steps {
        script{
            withDockerRegistry(credentialsId:'Docker_cred',url: https://index.docker.io/v1/){ sh 'docker push '
            }
        }
    }
}

stage('Deploy
    App'){ steps
    {
        withKubeConfig(caCertificate:'',clusterName:'minikube',contextName:'minikube',
credentialsId:'mukubeconfig_011',namespace:'',restrictKubeConfigAccess:false,serverUrl:
        '') {
            sh'kubectlapply -fdeployment.yml --validate=false'
        }
    }
}

stage('Test'){ steps {

```

```
withKubeConfig(caCertificate: "", clusterName: 'minikube', contextName: 'minikube',
credentialsId:'mukubeconfig_011',namespace:"",restrictKubeConfigAccess:false,serverUrl:
    ') {
```

```
    sh'minikubeservice-my-service--url|xargscurl'
```

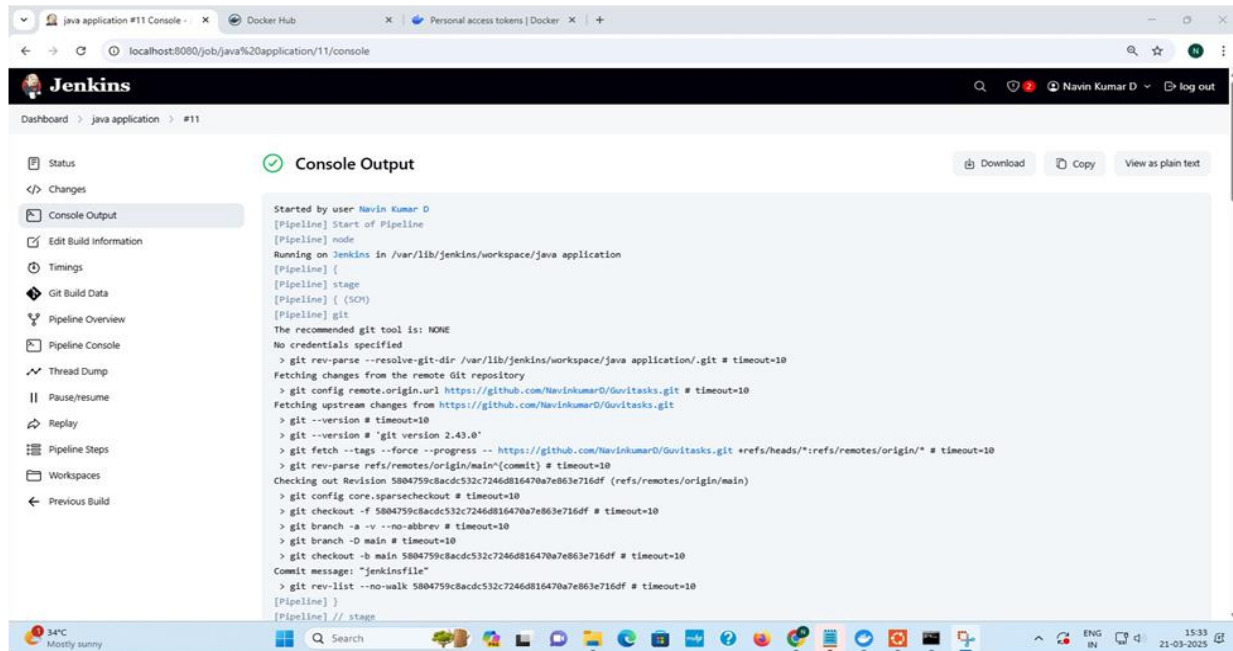
```
}
```

```
}
```

```
}
```

```
}
```

```
}
```



Deployment.yml

```

apiVersion:apps/v1 kind:Deployment
metadata:
  name: my-deploy
  namespace:my-y-bank
  labels:
    name:my-deploy
spec:
  replicas:1
  selector:
    matchLabels:
      apptype:web-backend
  strategy:
    type:RollingUpdate
  template:
    metadata:
      labels:
        apptype:web-backend
    spec:
  
```

```

    containers:
      - name: my-web
        image: 22csl258/mysimplewebapplication:tagname
        ports:
          - containerPort: 9001
  apiVersion:
v1 kind:
Service
metadata:
  name: my-
  service
  namespace: my-
  y-bank
  labels:
    app: my-
  service spec:
    type: NodePort
    ports:
      - port: 9001
        targetPort: 8080

```

Service.yml

```

apiVersion: apps/v1
kind: Deployment
metadata:
  name: my-deploy
  labels:
    name: my-deploy
spec:
  replicas: 1
  selector:
    matchLabels:
      apptype: web-backend
  strategy:
    type: RollingUpdate
  template:
    metadata:
      labels:
        apptype: web-backend
    spec:
      containers:
        - name: my-app
          image: 22csl258/simplewebapp:latest
          ports:

```

- containerPort: 9000

apiVersion: v1

kind: Service

metadata:

name: my-service

labels:

app: my-service

spec:

type: NodePort

ports:

- port: 9000

targetPort: 8080

nodePort: 30002

selector:

apptype: web-backend

```
navin@ITP-CC16-42: ~  
❯ The control-plane node minikube host is not running: state=Stopped  
❯ To start a cluster, run: "minikube start"  
navin@ITP-CC16-42:~$ minikube start  
❯ minikube v1.35.0 on Ubuntu 24.04 (amd64)  
* Using the docker driver based on existing profile  
❯ Starting "minikube" primary control-plane node in "minikube" cluster  
❯ Pulling base image v0.0.46 ...  
❯ Restarting existing docker container for "minikube" ...  
❯ Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...  
❯ Verifying Kubernetes components...  
  * Using image gcr.io/k8s-minikube/storage-provisioner:v5  
  * Enabled addons: storage-provisioner, default-storageclass  
  * Done! kubect1 is now configured to use "minikube" cluster and "default" namespace by default  
navin@ITP-CC16-42:~$ minikube service my-service  
-----  
| NAMESPACE | NAME      | TARGET PORT | URL                               |  
-----  
| default   | my-service | 9000         | http://192.168.49.2:30002       |  
-----  
* Starting tunnel for service my-service.  
-----  
| NAMESPACE | NAME      | TARGET PORT | URL                               |  
-----  
| default   | my-service |             | http://127.0.0.1:44875         |  
-----  
* Opening service default/my-service in default browser...  
❯ http://127.0.0.1:44875  
! Because you are using a Docker driver on linux, the terminal needs to be open to run it.  
^C❯ Stopping tunnel for service my-service.  
navin@ITP-CC16-42:~$ minikube service my-service  
-----  
| NAMESPACE | NAME      | TARGET PORT | URL                               |  
-----  
| default   | my-service | 9000         | http://192.168.49.2:30002       |  
-----  
* Starting tunnel for service my-service.  
-----  
| NAMESPACE | NAME      | TARGET PORT | URL                               |  
-----  
| default   | my-service |             | http://127.0.0.1:45581         |  
-----  
* Opening service default/my-service in default browser...  
❯ http://127.0.0.1:45581  
! Because you are using a Docker driver on linux, the terminal needs to be open to run it.
```